



[12] 发明专利申请公布说明书

[21] 申请号 200580025147.5

[43] 公开日 2007年6月27日

[11] 公开号 CN 1989478A

[22] 申请日 2005.7.8
 [21] 申请号 200580025147.5
 [30] 优先权
 [32] 2004.7.26 [33] US [31] 10/899,530
 [86] 国际申请 PCT/US2005/024375 2005.7.8
 [87] 国际公布 WO2006/019636 英 2006.2.23
 [85] 进入国家阶段日期 2007.1.25
 [71] 申请人 英特尔公司
 地址 美国加利福尼亚州
 [72] 发明人 E·萨姆森 A·纳瓦勒
 L·克莱恩

[74] 专利代理机构 上海专利商标事务所有限公司
 代理人 陈 斌

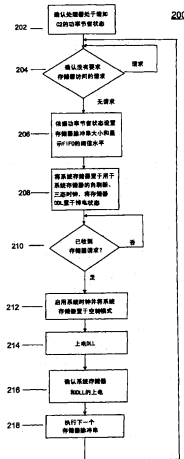
权利要求书 4 页 说明书 8 页 附图 5 页

[54] 发明名称

用于动态 DLL 掉电和存储器自刷新的方法和装置

[57] 摘要

本发明的实施例提供了一种用于节省电子设备内功率的方法和装置。更具体地，本发明的实施例动态地将存储器置于自刷新状态并将芯片组时钟电路置于掉电模式，同时还在功率节省模式下保持等时数据流（例如，显示）的更新并服务总线主控周期。



1. 一种用于节省电子设备内功率的方法，包括：
响应于没有未完成的存储器请求而自动将所述电子设备转换至操作的功率降低模式。
2. 如权利要求 1 所述的方法，其特征在于，还包括：
响应于确定性的一组配置被满足而自动将所述电子设备转换至操作的功率降低模式。
3. 如权利要求 2 所述的方法，其特征在于，响应于确定性的一组配置被满足而自动将所述电子设备转换至操作的功率降低模式还包括：
响应于确定性的一组配置被满足而将所述存储器置于自刷新状态。
4. 如权利要求 3 所述的方法，其特征在于，响应于确定性的一组配置被满足而自动将所述电子设备转换至操作的功率降低模式还包括：
响应于确定性的一组配置被满足而将时钟、控制信号、时钟树、DLL 或其他不必要的逻辑/电路置于掉电模式。
5. 如权利要求 4 所述的方法，其特征在于，响应于确定性的一组配置被满足而自动将所述电子设备转换至操作的功率降低模式还包括：
在所述降低功率模式下保持所述等时数据的更新及服务总线主数据。
6. 如权利要求 5 所述的方法，其特征在于，所述功率节省模式包括 C2 功率节省模式。
7. 如权利要求 5 所述的方法，其特征在于，响应于确定性的一组配置被满足而将所述存储器置于自刷新状态还包括：
确定所述等时数据和总线主数据的组合是否超过预定缓冲阈值；以及
响应于所述组合没有超过预定阈值而将所述存储器置于自刷新状态。
8. 如权利要求 7 所述的方法，其特征在于，所述预定阈值包含存储器离开自刷新的最大退出等待时间。
9. 如权利要求 8 所述的方法，其特征在于，等时数据包括显示数据。
10. 如权利要求 8 所述的方法，其特征在于，确定所述等时数据和总线主数据的组合是否超过预定缓冲阈值还包括：
访问等时数据和总线主数据的参数；以及

使用参数预计算掉电模式退出等待时间是否位于所述预定阈值内。

11. 如权利要求 10 所述的方法，其特征在于，访问等时数据和总线主数据的参数还包括：

使用 bios/驱动器来访问等时数据和总线主数据的参数。

12. 如权利要求 11 所述的方法，其特征在于，还包括：

通过编码存储器控制器配置寄存器或编码控制诸如存储器自刷新或 DLL 掉电、或时钟禁用的掉电模式的状态机来表示所述计算。

13. 如权利要求 12 所述的方法，其特征在于，还包括：

在运行中计算掉电退出等待时间是否位于所述预定阈值之内。

14. 如权利要求 8 所述的方法，其特征在于，确定所述等时数据和总线主数据的组合是否超过预定阈值还包括：

计算所述最大掉电退出时间，该计算依据：

最大掉电退出时间 = 自刷新退出时间 + 退出时间实现开销/无效率 + DLL 掉电退出时间的适用分数。

15. 如权利要求 14 所述的方法，其特征在于，显示等待时间容限是依据 FIFO 大小和显示模式要求确定的。

16. 如权利要求 15 所述的方法，其特征在于，显示等待时间容限大于所述最大掉电退出时间。

17. 如权利要求 16 所述的方法，其特征在于，等时等待时间容限是由 FIFO 大小和最小周期性间隔要求所确定的。

18. 如权利要求 17 所述的方法，其特征在于，所述等时等待时间容限大于所述最大掉电退出时间。

19. 一种系统，包括：

存储器，以及

电源管理逻辑，用于响应没有未完成的存储器请求而自动将电子设备转换至操作的功率降低模式。

20. 如权利要求 19 所述的系统，其特征在于，所述电源管理逻辑响应于确定性的一组配置被满足而自动将所述电子设备转换至操作的功率降低模式。

21. 如权利要求 20 所述的系统，其特征在于，所述电源管理逻辑响应于

确定性的一组配置被满足而将所述存储器置于自刷新状态。

22. 如权利要求 21 所述的系统，其特征在于，所述电源管理逻辑响应于确定性的一组配置被满足而将时钟或 DLL 置于掉电模式。

23. 如权利要求 22 所述的系统，其特征在于，所述电源管理逻辑在降低功率模式下保持等时数据的更新及服务总线主数据。

24. 如权利要求 23 所述的系统，其特征在于，所述功率节省模式包括 C2 功率节省模式。

25. 如权利要求 23 所述的系统，其特征在于，所述电源管理逻辑确定所述等时数据和总线主数据的组合是否超过预定缓冲阈值，并且响应于所述组合没有超过预定阈值而将所述存储器置于自刷新状态。

26. 如权利要求 25 所述的系统，其特征在于，所述预定阈值包含存储器离开自刷新的所述最大退出等待时间。

27. 如权利要求 26 所述的系统，其特征在于，等时数据包括显示数据。

28. 如权利要求 26 所述的系统，其特征在于，所述电源管理逻辑访问等时数据和总线主数据的参数，并且使用参数预计算掉电模式退出等待时间是否位于所述预定阈值内。

29. 如权利要求 28 所述的系统，其特征在于，所述电源管理逻辑使用 bios 或驱动器来访问等时数据和总线主数据的参数。

30. 如权利要求 25 所述的系统，其特征在于，所述电源管理逻辑在运行中计算掉电退出等待时间是否位于所述预定阈值之内。

31. 一种包括指令的机器可访问介质，所述指令被执行时会导致机器：响应于没有存储器请求而自动将电子设备转换至操作的功率降低模式。

32. 如权利要求 31 所述的机器可访问介质，其特征在于，还包括：响应于确定性的一组配置被满足而将所述电子设备转换至操作的功率降低模式。

33. 如权利要求 31 所述的机器可访问介质，其特征在于，响应于确定性的一组配置被满足而将所述电子设备转换至操作的功率降低模式还包括：

响应于确定性的一组配置被满足而将所述存储器置于自刷新状态。

34. 如权利要求 31 所述的机器可访问介质，其特征在于，响应于确定性

的一组配置被满足而将所述电子设备转换至操作的功率降低模式还包括：

响应于确定性的一组配置被满足而将时钟、控制信号、时钟树、DLL 或其他不必要的逻辑/电路置于掉电模式。

35. 一种系统，包括：

存储器管理器，用于响应没有未完成的存储器请求而自动将所述电子设备转换至操作的功率降低模式。

36. 如权利要求 35 所述的系统，其特征在于，所述存储器管理器响应于确定性的一组配置被满足而将所述电子设备转换至操作的功率降低模式。

37. 如权利要求 35 所述的系统，其特征在于，所述存储器管理器响应于确定性的一组配置被满足而将时钟或 DLL 置于掉电模式。

38. 如权利要求 35 所述的系统，其特征在于，所述存储器管理器在降低功率模式下保持等时数据的更新，及服务总线主数据。

用于动态 DLL 掉电和存储器自刷新的方法和装置

背景

计算设备，尤其是便携设备经常受到其重新连接到 AC 电源之前能依靠电池电源运行时间的限制。这就使得人们持续为降低包括中央处理单元在内的计算机部件的功耗做出努力。将诸如中央处理单元、存储器控制器或存储器的电子设备保持在它们尽可能低的功率状态可以提供不少益处。例如，它允许用电池的机器在重新充电之间更长时间的运行。功耗的降低还能降低中央处理单元的热耗散。热耗散的降低允许中央处理单元在全速运行更长时间的同时，保持其热耗散的技术要求。热耗散的降低还能减轻对风扇以及用于防止计算机产生热量的其他部件的需求。

用于开发电源管理系统的标准规范是高级配置和电源接口 (ACPI) 规范 (例如，2000 年 7 月 27 日发布的版本 2.0；还可参见于 2001 年 2 月 27 日发布的版本 1.05 的 ACPI Component Architecture Programmer Reference，该参考可从加利福尼亚州圣克拉拉市的 Intel 公司获取)。ACPI 的一个目标是增强电源管理功能和稳健性，并且方便公共电源管理特征的广泛工业应用。

ACPI 定义了作为在全局工作状态下的处理器功耗和热管理状态的多个处理器功率状态。这些处理器状态包括 (i) C0 功率状态、(ii) C1 功率状态、(iii) C2 功率状态以及(iv) C3 功率状态。在 C0 功率状态中，处理器执行指令并以全功率运行。在 C1 和 C2 功率状态中，处理器是非执行功率状态。然而，C2 功率状态使用的功率要小于 C1 状态。在 C1 和 C2 功率状态中，处理器仍然允许总线窥探处理器高速缓存并由此维持高速缓存连贯性。C3 功率状态比 C1 和 C2 功率状态更节省功率，但这是以对存储器更长的掉电退出等待时间为代价的。

在传统系统中，电源管理逻辑在某些环境下引发 CPU 从 C2 功率状态转换回到较高功率的 C0 功率状态。将电子设备保持在比其他情况能够达到的更低的功率状态，并降低功率状态之间的转换次数就能够通过降低由在设计的功率状态之间切换所引起的等待时间同时保持整个功耗更低来改进系统性能。

附图简述

图 1 示出了在 ACPI 规范下各处理器功率状态之间转换的实施例的示意图。

图 2 示出了在集成图形配置下用于将存储器置于自刷新和将存储器数字锁定环路 (DLL) 置于掉电模式, 同时在 C2 功率状态期间保持显示器更新并维持总线主控的使用的过程的实施例的流程图。

图 3 示出了用于将存储器置于自刷新并将 DLL 置于掉电模式, 同时在 C2 功率状态期间维持总线主控的使用并保持显示器的更新的示例性集成图形配置的实施例的示意图。

图 4(a)和 4(b)示出了在分立配置下用于将存储器置于自刷新并将 DLL 置于掉电模式, 同时维持在 C2 功率状态期间总线主控使用的过程的实施例的流程图。

详细描述

本发明的实施例提供了一种用于保持电子设备内功率的方法和装置。更具体地, 本发明的实施例动态地将存储器置于自刷新并将芯片组时钟电路置于掉电模式, 同时保持显示器的更新并在诸如 C2 的功率节省模式中服务总线主控周期。将处理器维持在诸如 C2 的功率节省模式在即使出现可窥探总线主控周期的情况下 (例如, 与 C3 状态不同) 仍能节省功率并降低集成和非集成图形芯片组平台之间的功率差。

在随后的描述中将阐明多个细节以提供对本发明的透彻理解。然而本领域的普通技术人员应该理解无需这些具体细节也能实现本发明。在其他实例中, 将不会详细描述公知的方法、过程、部件和电路结构和设备以免淡化本发明的主旨。

将按照计算机内对数据位或二进制数字信号操作的算法和符号表示在随后的详细描述中的某些部分。这些算法描述和表示是数据处理技术领域的普通技术人员用来将他们的工作主旨传达给该领域其他普通技术人员的方法。在此的算法通常被认为是可导致期望结果的自我协调的步骤序列。这些步骤包括对物理量的物理处理。在通常但非必须的情况下, 这些量具有能被存储、传输、组合、比较以及其他处理的电或磁信号形式。已证实有时为了便于广泛使用, 可将这些信号称为比特、值、元素、符号、字符、项、数字等等。但应该理解所有这些及类似的术语都与合适的物理量相关联并且只是应用于这些物理量的方便的标记。

除非另外特别指出否则如可以从以下讨论中显见,可以理解贯穿本说明书使用诸如“处理”、“计算”、“演算”或“确定”之类的术语的讨论指的是计算机或计算系统或者类似的数据处理设备的动作和进程,这些设备将计算机系统的寄存器和/或存储器内被表示为物理(例如,电子)量的数据处理并变换成计算机系统存储器或寄存器或其他这类信息存储、传输或显示设备内类似地被表示为物理量的其他数据。

本发明的实施例可由硬件或软件,或两者的组合实现。然而本发明的实施例可以被实现为在包括至少一个处理器、数据存储系统(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备和至少一个输出设备的可编程系统上执行的计算机程序。可以对输入数据应用程序代码以执行在此描述的功能并生成输出信息。可以用已知的方式将输出信号应用于一个或多个输出设备。为了这一应用,处理系统可以包括任何带有,例如数字信号处理器(DSP)、微控制器、专用集成电路(ASIC)或微处理器的处理器的系统。

各程序可由高级过程型或面向对象的编程语言来实现以与处理单元通信。各程序在需要时还可由汇编或机器语言实现。实际上,本发明不限于任何特定的编程语言。在任何情况下,该语言可以是被编译的或是解释语言。

各程序可以存储在通用或专用可编程处理系统可读的存储介质或设备(例如,硬盘驱动器、磁盘驱动器、只读存储器(ROM)、CD-ROM设备、闪存设备、数字视频盘(DVD)或其他存储设备)中,这些程序用于在该存储介质或设备被处理系统读取时配置并操作该处理系统以执行在此描述的过程。可以考虑将本发明的实施例实现为被配置与处理系统一并使用的机器可读存储介质,其中那样配置的存储介质使得处理系统以特定和预定的方式操作以执行在此描述的功能。

图1示出了在ACPI规范下各处理器功率状态之间转换的实施例100的示意图。所有的状态,即C0状态102、C1状态104、C2状态106和C3状态108都被包括在G0工作状态110中。G0工作状态被ACPI规范定义为其中系统分派用户模式(应用)线程的计算机状态。在G0工作状态中,这些线程都被执行。在该状态中,设备(外围设备)的功率状态动态变化。在G0状态110内,处理器在各处理器功率状态之间转换,包括C0状态102 C1状态104、C2状态106和C3状态108。

在C0状态102中,处理器处于全功率。在此状态下,典型系统的各部件都有

电源提供并且该系统内的时钟能够全速运行。C1 状态 104 定义了在其中处理器功率状态具有最低等待时间的非执行状态。

C2 状态 106 是比 C1 状态 104 更省电的第二非执行功率状态。C2 状态 106 是共用芯片组模式，而计算机则处于无源状态（即，操作系统空转）并连接至诸如 USB 设备或音频端口的总线主控。在 C2 状态 106 期间，分立芯片组访问主要服务总线主控周期的存储器而集成图形芯片组访问主要获取显示器刷新数据、服务总线主控周期或继续图形渲染的存储器。CPU 无需访问存储器。DRAM 存储器在有时被称为备用模式或自刷新的扩展功率保存模式下工作。刷新单元重新充电 DRAM 存储器内的电学单元以维持数据的完整。

C3 功率状态 108 提供比 C1 状态 104 和 C2 状态 106 都更为改进的功率节省效果。当处于 C3 状态 108 时，处理器的高速缓存维持当前信息状态并且无法窥探。处理器会被带回 C0、C1 或 C2 状态以处理可窥探的通信量。

状态之间的转换发生在从 C0 状态 102 沿着路径 112 至 C1 状态 104 并沿着返回路径 114 返回到 C0 状态 102。状态间的转换还发生在从 C0 状态 102 沿着路径 116 至 C2 状态 106 并沿着返回路径 118 返回到 C0 状态 102。最后，状态间的转换还发生在从 C0 状态 102 沿着路径 120 至 C3 状态 108 并沿着返回路径 122 返回到 C0 状态 102。CPU 在足够长的时间内持续不活动将引起从 C0 状态 102 沿着路径 116 至 C2 状态 106 的转换。诸如中断的打断事件将会导致系统从 C2 状态 106 沿着路径 118 转换回 C0 状态 102。

应该认识到虽然为了方便将根据 C0、C1、C2 和 C3 的 ACPI 规范功率状态来描述该系统，但是本发明不受 ACPI 规范的限制。通常对于不遵守 ACPI 规范的实施例来说，可以出于本发明的目的将 C0 功率状态定义为在其中 CPU 执行正常功能的全功率状态。ACPI C2 功率状态通常被定义为全功率和 C3 功率状态之间的中间功率状态。对于 Intel 处理器来说，C2 功率状态等效于 STOP GRANT 状态。通常 C2 功率状态允许窥探存储器访问并维持高速缓存连贯性。

图 2 示出了在集成图形配置下用于将存储器置于自刷新状态和将存储器数字锁定环路（DLL）置于掉电模式，同时在 C2 功率状态期间保持显示器更新并维持总线主控的使用的过程的实施例 200 的流程图。本发明的实施例（1）在空转期间将存储器置于自刷新状态，而不是仅仅置于预充电掉电模式和/或（2）动态掉电

DDR 时钟/DLL。出于本发明的目的，该功率节省状态即使在获取的功率节省效果比存储器仅进行自刷新更好的情况下仍被称为“C2 自刷新”。更具体地，因为在该平台上的其他总线主控与显示器相比具有更大的等待时间容限，所以只要为显示器提供的缓冲足以覆盖用于存储器离开自刷新的最大退出等待时间就能够合适进行显示器更新。如果在必须服务显示器请求时不等时总线主控已经开始向存储器传输极长的脉冲串，那么就延长总线主控的请求直到该显示器请求服务之后。只要也必须进行存储器访问的任何同步数据流（例如，同步音频）的脉冲串大小短到足以位于其他同步数据流（例如，显示）等待时间处理能力之内，并且只要这些数据流以低于退出存储器自刷新所需的速率来请求存储器访问，则随后就能启用 C2 自刷新状态。因为同步数据流的最大脉冲串大小和最小重复频率在平台内具有确定性这一特征，所以就能轻易得知何时可达到 C2 自刷新状态。

在步骤 202 中，确认处理器处于 C2 功率状态。

在步骤 204 中，确认没有来自任何源（总线主控、显示器刷新）的存储器请求。

在步骤 206 中，将存储器脉冲串大小和显示 FIFO 阈值水平设置为适应于 C2 功率状态的预定水平。更具体地，如图 3 和图 4 所示并在随后将详述的那样，显示器 FIFO 具有在达到时会触发脉冲串请求的阈值水平。设置 FIFO 的阈值水平使得显示器刷新所需的存储器脉冲串足够长并且间隔时间足够长，这样在需要重新启用 DDR DLL 和芯片组存储器之前 C2 功率状态下的较大的掉电时间就是有可能的。在用于集成图形配置的典型配置中，显示逻辑管理显示 FIFO。在阈值寄存器中寄存该阈值。该阈值是可编程的并且可以依据功率节省模式预设。这样就能通过限制存储器传输次数（每次都消耗功率的）来节省功率，并且能够在低功率设备进入功率节省模式的静态显示期间产生空转周期。请求的脉冲串大小和阈值水平控制这些请求的时间间隔。

确认或迫使渲染引擎空转。当不要求或完成图形渲染时，芯片组通常处于能够提供用于进入自刷新状态机会的状态。

在步骤 208 中，会发生如下情况的任意一种或这些情况的组合：1) 将系统存储器置于带有用于该系统存储器的三态时钟和其他存储器控制信号的自刷新状态，2) 能够将在 C2 自刷新状态期间不需要的存储器 DLL 置于掉电和/或 3) 能够将在

C2 自刷新状态期间不需要的任何其他功能块和时钟树置于掉电状态。决定哪些功能掉电取决于包括比较掉电特征的掉电退出等待时间相对于可用时间的影响的决定逻辑。可用时间则取决于显示器可以忍受的最大等待时间、等时数据流的周期性和脉冲串大小的要求。

可以将存储器 DLL 置于掉电模式。更具体地，诸如 DDR DRAM 的集成电路通常生成多个同步的 DLL 输出（相位）并且利用多个操作模式，使得由诸如 DLL 的电路所产生的输出信号被选择性地应用于该设备中的电路以降低不必要的功耗。在典型的实现中，电源管理单元控制为在诸如处理器、存储器控制器和存储器的系统内其他芯片提供时钟信号的时钟发生器。诸如 DDR DRAM 的集成电路通常包括为多个电路提供分布信号（例如，时钟信号）的 DLL。DLL 通常接收参考时钟信号，从参考时钟信号生成内部时钟信号，而该内部时钟信号的相位通常取决于参考时钟信号。DLL 比较复杂并且在高频下工作，因而消耗大量功率。希望用这一内部时钟信号同步的操作多个电路。如果这些电路被共同驱动，那么加在 DLL 上的总输出负载会非常大，从而引起 DLL 消耗大量的功率。因此掉电 DLL 是很有利的。

在步骤 210 中，保持自刷新和动态 DLL 的掉电状态，直到确认总线主控请求和/或显示器刷新。

在步骤 212 中，响应于对总线主控和/或显示器刷新已经被执行的确认，启用系统存储器时钟并将系统存储器置于空转模式。

在步骤 214 中，对 DLL 上电。在 C2 状态期间，可选地将与用于更新显示器刷新的存储器相关联的芯片组 DLL 保持在启用状态。

在步骤 216 中，系统等待，直到 DLL 和系统存储器均上电。

在步骤 218 中，执行下一个存储器脉冲串并且该过程返回步骤 204。只要没有打断事件（例如，一中断），则处理器保持在 C2 功率状态。

在典型实现中，处理器时钟被重新启动或被发信号至未赋值的处理器以完成该转换。随后则依据 C0 功率状态要求来设置存储器脉冲串大小和水印水平。在诸如 C0 的全功率状态操作期间，存储器脉冲串通常更小并且依据 C0 功率状态而间隔更短的时间。C0 状态利用了大到足以包容新的 C2 脉冲串大小和本发明阈值水平要求的显示 FIFO 大小。

以上处理总线请求同时让处理器处于低功率状态的方法可由如下将详述的各

种不同装置实现。

例如，图3是用于将存储器置于自刷新并将DLL置于掉电模式，同时如图2所示在C2功率状态期间维持总线主控的使用并保持显示器的更新的示例性集成图形配置的实施例的示意图。计算机系统300包括处理器302、包括图形引擎306的图形和存储器控制器304、存储器308、显示FIFO310、显示流水线312和显示设备314。处理器302处理数据信号并且可以是复杂指令集计算机（CISC）微处理器、精简指令集计算机（RISC）微处理器、超长指令字（VLIW）微处理器、实现指令集组合的过程或其他处理设备，诸如数字信号处理器。处理器302可以耦合至在处理器302和系统300内其他部件之间传输数据信号的公共总线312。

处理器302将信号发至公共总线312用来与存储器308或图形和存储器控制器304通信以处理在此描述的数据。处理器302发出这些信号来响应于从存储器308中获取的软件指令。存储器308可以是动态随机存取存储器（DRAM）设备、静态随机存取存储器（SRAM）设备，或者其他存储器设备。存储器308可以存储可由处理器302、图形引擎306或其他设备执行的指令和/或由数据信号表示的数据。指令和/或数据可以包括用于执行本发明任何和/或全部技术的代码。存储器308还可以含有软件和/或数据。可选的高速缓存可用于加速由图形引擎306利用其访问局部性的优势而进行的存储器访问。在某些实施例中，图形引擎306可以从处理器302中卸载许多渲染图像所需的存储器密集型任务。图形引擎306处理数据信号并且可以是复杂指令集计算机（CISC）微处理器、精简指令集计算机（RISC）微处理器、超长指令字（VLIW）微处理器、实现指令集组合的过程或其他处理设备，诸如数字信号处理器。图形引擎306可以耦合至在图形引擎306和系统300内包括渲染高速缓存310和显示设备314等其他部件之间传输数据信号的公共总线312。图形引擎306包括尤其可以将特定属性（例如，颜色）写入显示器314特定像素并在显示设备314上绘制复杂图元的渲染硬件。图形和存储器控制器304与显示设备314通信，用于由图形控制器304为了用户显示渲染或其他处理的图像而显示经渲染或处理的图像。显示设备314可以包括计算机监视器、电视机、平板显示器或其他合适的显示设备。

存储器308存储可以包括一个或多个渲染程序来建立用于显示的图像图元的图像的主机操作系统。系统300包括图形引擎306，诸如使用专用硬件逻辑设备或

协处理器来改进由主机渲染程序处理的渲染至少部分图形图元的性能的图形加速器。主机操作系统程序和它的主机图形应用程序接口（API）通过驱动器程序控制图形引擎 306。

FIFO 310 通过数据总线 318 接收来自图形和存储器控制器 304 的显示数据并将显示数据通过数据总线 320 输出至显示流水线 312。图形和存储器控制器 304 确定哪个设备应该被准许访问存储器 308。图形引擎的一部分控制图像传入、传出存储器 308 或在其内的块传输。存储器地址发生器 322 连接至图形和存储器控制器 304 和显示 FIFO 310。存储器地址发生器 322 生成存储器地址并将其送入图形和存储器控制器 304。图形和存储器控制器 304 控制存储器地址发生器 322 和显示流水线 312。图形和存储器控制器 304 在开始载入 FIFO 310 时发指令给存储器地址发生器 322。显示 FIFO 310 可用于接收并存储用于显示设备 314 的显示数据。

当 FIFO 水平大于阈值时，就可以在不损害显示的情况下生成用于非显示数据流的存储器脉冲串请求。基于 FIFO 数据水平与阈值的比较，控制电路发送请求给图形和存储器控制器 304 用于存储器访问，这样就能如图 1 中流程图所示将数据载入 FIFO 310。

图 4(a)和 4(b)示出了在分立配置下用于将存储器置于自刷新并将 DLL 置于掉电模式，同时在 C2 功率状态期间维持总线主控使用的过程的实施例的流程图。分立的芯片组配置不具有图形，并且只要满足等时限制（即，等时的周期性必须大于掉电退出等待时间）就能让存储器处于自刷新状态。分立的图形控制器具有需要维持的显示数据流。但是分立的图形控制器对 C2 状态一无所知。

参见图 4(a)，在一个实施例 400 中，只要没有对本地存储器的未完成请求（步骤 402），分立的图形控制器就进入它与诸如自刷新状态的掉电模式（出于参考的目的，被称为图形 C2 功率状态）相关的本地存储器（步骤 404）。

参见图 4(b)，在另一实施例 406 中，分立的图形控制器基于带宽阈值和/或在本地存储器上该本地存储器请求的空转持续时间来计算要求（步骤 408）。响应于该足够低的要求，它就进入让其本地存储器进入自刷新（步骤 410）。

已经依据专利法令的要求描述了本发明，本领域普通技术人员将理解如何对本发明做出变化和修改以达到他们特定的要求或条件。做出的这些变化和修改不背离由所附权利要求阐明的本发明的范围和精神。

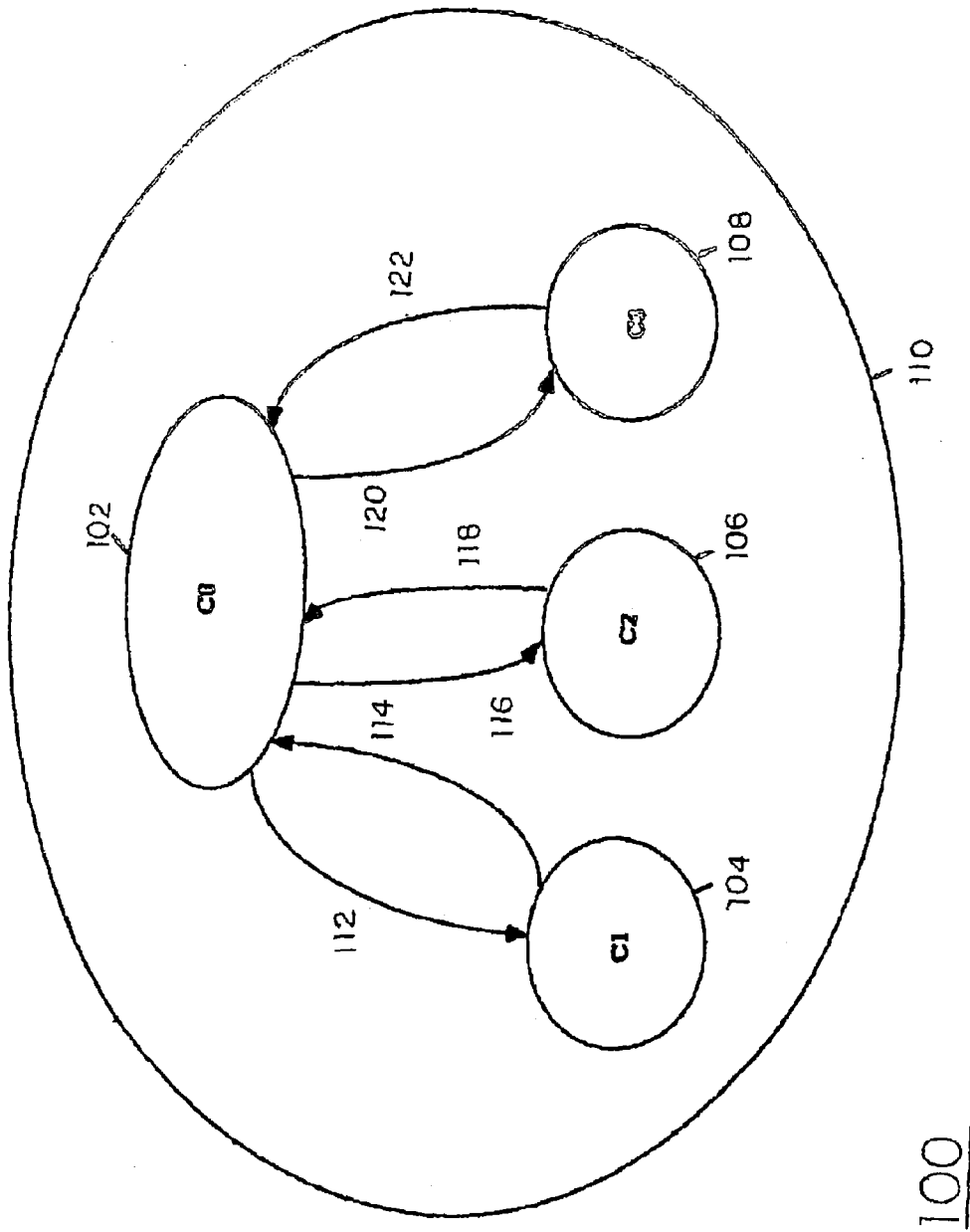


图 1

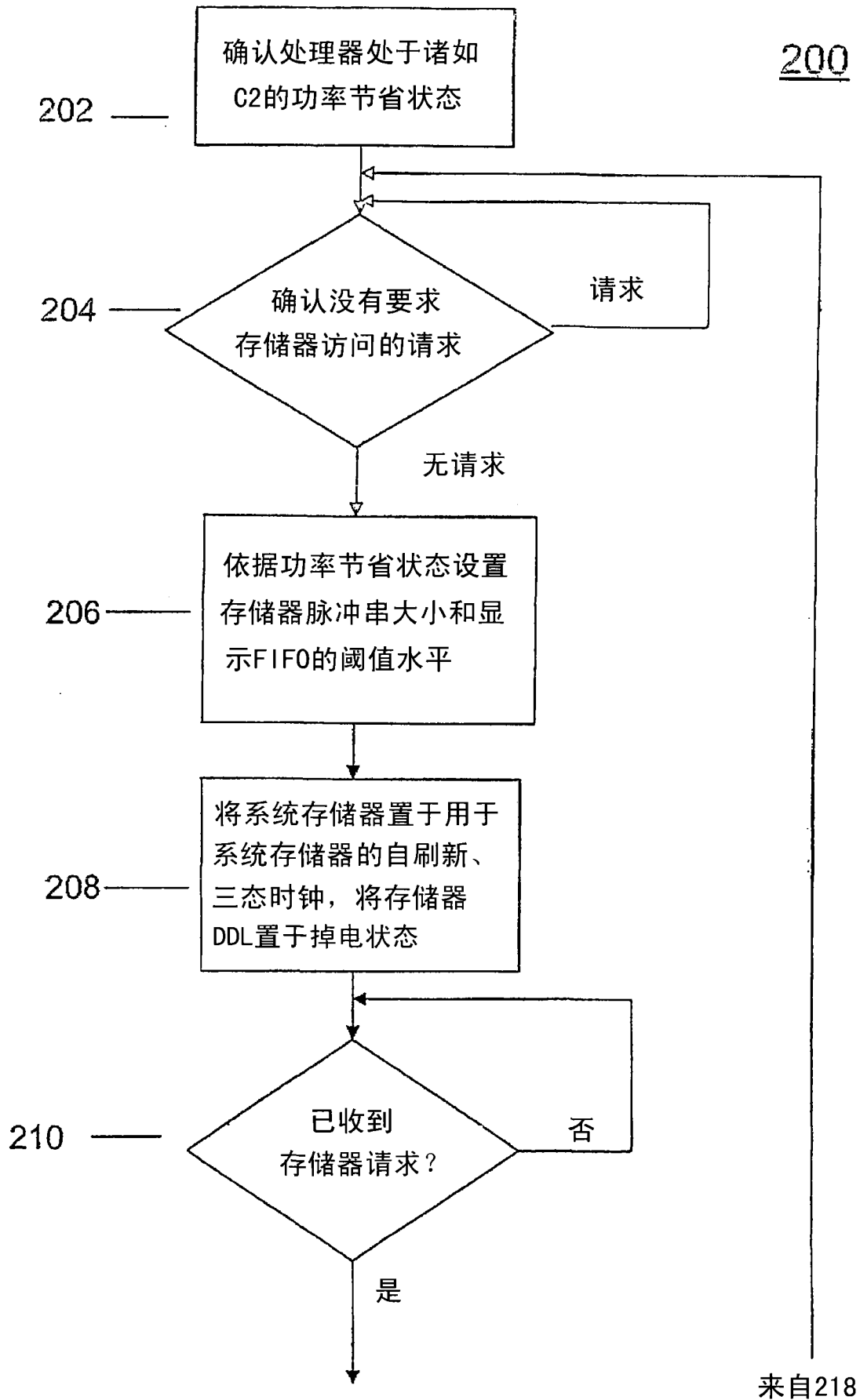


图 2

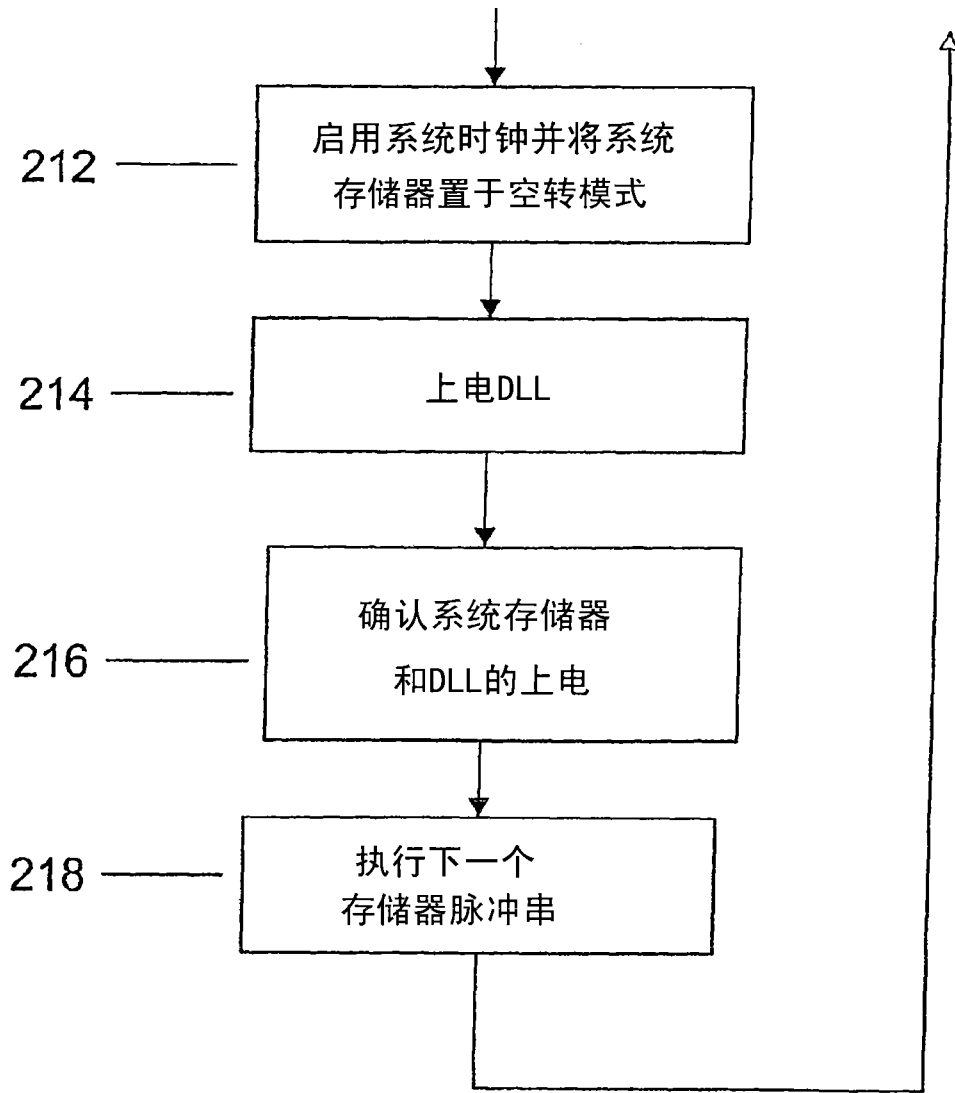


图 2(续)

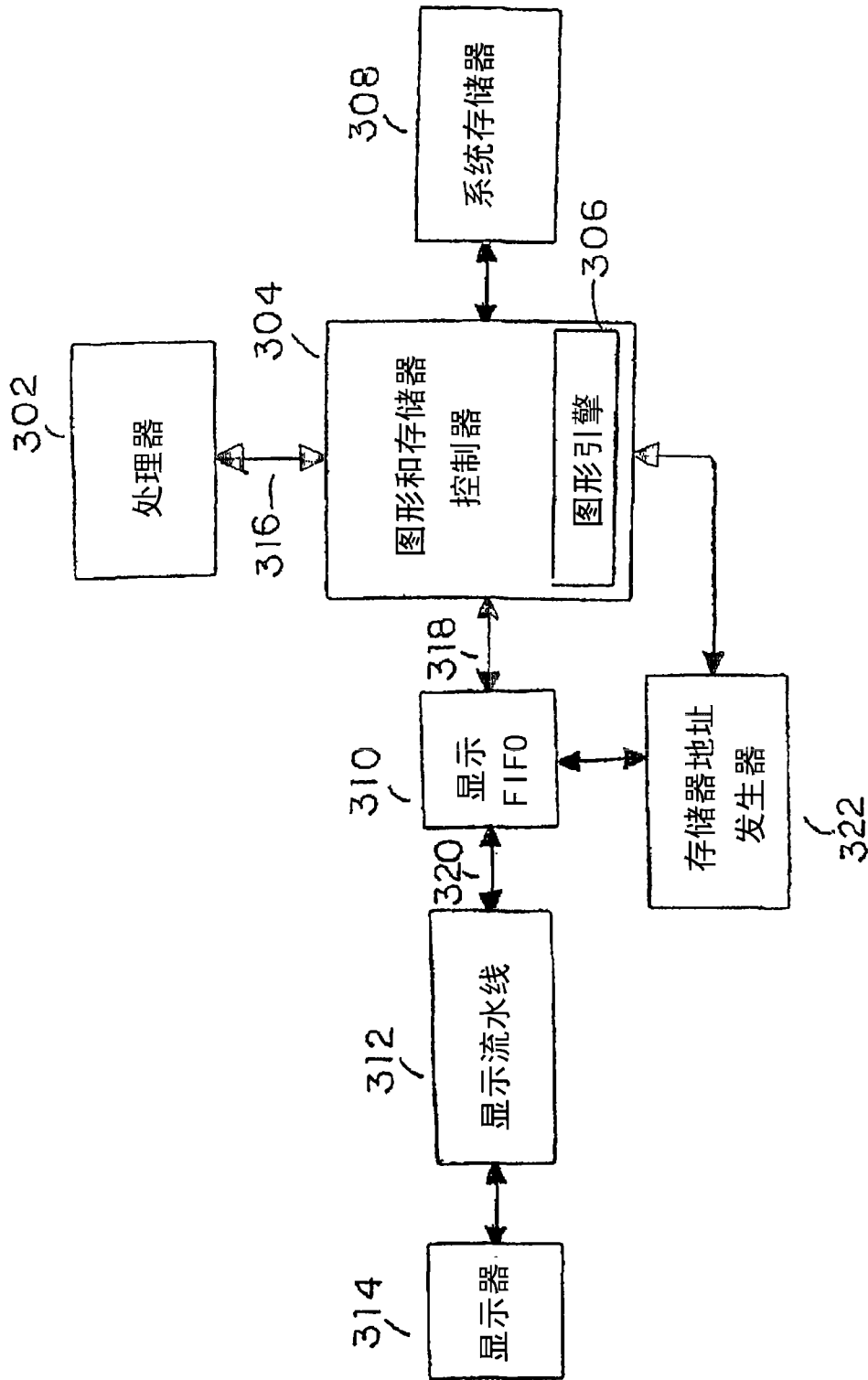


图 3

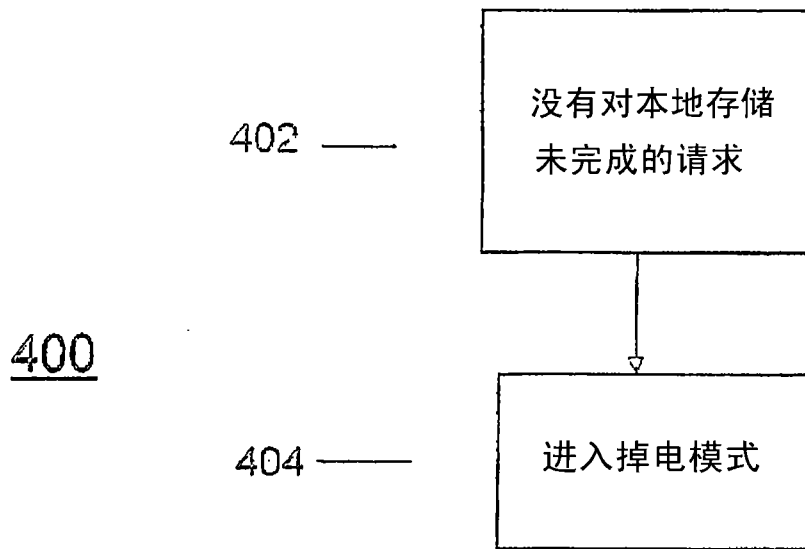


图 4(a)

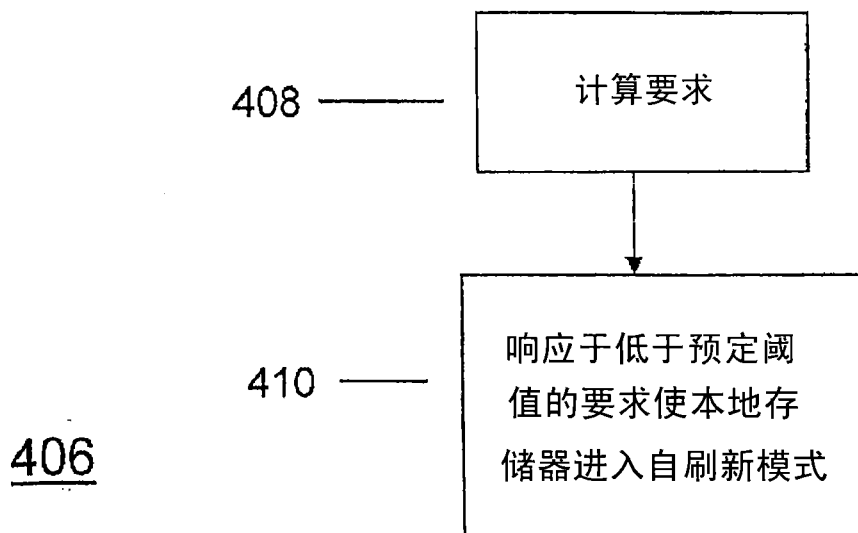


图 4(b)